

## Research Article

# Study about Law Multi-Issue Automatic Negotiation Method Based on Artificial Intelligence and Multiagent Evolutionary Algorithm

Yu-Ting Hsu<sup>1</sup> and Cheng-Yong Liu <sup>2</sup>

<sup>1</sup>The Department of Ocean and Border Governance, National Quemoy University, Kinmen County 892009, Taiwan

<sup>2</sup>School of Marine Culture and Law, Jimei University, Xiamen, Fujian 361001, China

Correspondence should be addressed to Cheng-Yong Liu; [cy.liu@jmu.edu.cn](mailto:cy.liu@jmu.edu.cn)

Received 19 July 2021; Accepted 23 August 2021; Published 6 September 2021

Academic Editor: Sang-Bing Tsai

Copyright © 2021 Yu-Ting Hsu and Cheng-Yong Liu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Multiagent System (MAS) is a self-learning intelligent system formed by many single agents. Each agent in the MAS works independently of the other, and they have all the characteristics of an agent system. It can respond to changes and countermeasures based on its own external environmental conditions. When solving a complex problem, multiple agents can form a group to solve the problem together. In this paper, the agent's evolutionary algorithm is integrated into the actual problem—multi-issue autonegotiation of law. According to this problem, the autonegotiation solution process and corresponding model are designed. In addition, a new type of solution is proposed for multiple legal issues. Compared with traditional solutions, the applicability has great advantages. Among them, the autonegotiation result of all the agent's total utility can be quickly found. In the changing environment, this article focuses on the multiagent system negotiation problem. According to the distributed information sharing of multiple agents, even if the case reveals incomplete information, the multiagent can be generated while ignoring the incomplete information. Optimal solution is proposed. The experimental results show that the success rate of the system in analyzing multiple legal issues and autonegotiations reached 67.56% under the condition of incomplete information from the outside world.

## 1. Introduction

Since the advent of artificial intelligence in 1956, after half a century of development, its research and application have gained a worldwide reputation in many fields. At present, presentation theory and technology are the hotspots of artificial intelligence technology research. A system composed of decentralized dynamic, large, and independent agents is called a multiagent system, but it also emphasizes independence, sociality, and the responsiveness of the entire system. A cooperative solution between multiagent systems is one of the core contents of multiagent system research. In order to complete the tasks set by multiple agents, they must interact with other agents to achieve their goals together.

This is because many agents lack an overall view, and there may be inconsistencies and conflicts in the goals,

procedures, knowledge, beliefs, and conclusions of many representative systems with world-class knowledge and global control. To achieve a consistent solution for the entire system, these inconsistencies or conflicts must be resolved. The best mechanism for resolving conflicts in a multiagent system is negotiation. Negotiation can be regarded as a way to resolve conflicts and coordinate cooperation, for example, by eliminating conflicts in planning objectives, resolving resource allocation constraints, clarifying work inconsistencies in the established organizational structure, and resolving inconsistencies. For many negotiation issues in the law, establishing a suitable multiagent negotiation mode and improving the bargaining power of the agent in the interaction process is one of the problems that must be solved by multiple parties. Self-negotiation involves a wide range of fields, such as e-commerce and business process

management. Supply chain conference planning involves a PCB assembly planning system, automatic negotiation, multiple legal issues, etc.

A multiagent system (MAS) is an intelligent system composed of multiple independent agents. SMA agents are independent of each other. With all the functions of agents, they can adapt to changes in the environment and solve related problems. At the same time, he can communicate with other agents and solve the same problem collaboratively. Bulling N proposed two new evolutionary algorithms. The improved ant colony algorithm provides a new optimized path for each path to avoid obstacles. The improved ant colony algorithm introduces a reverse learning method to start the distribution of ant positions, which improves the overall search ability of the algorithm. The particle swarm algorithm uses an adjustable inertia weight factor to adjust the Q value of the pheromone concentration, adjust the volatilization factor of the pheromone to adapt to changes and not drop to the optimal value, improve the iteration speed of the algorithm, use game theory to model dynamic obstacle avoidance among multiple agents, and use virtualization to solve the problem of Donash's game resolution and balanced decision. This algorithm is improved not only in speed but also in algorithm complexity, but the algorithm is being updated all over the world, and it has been difficult to keep up with the rhythm of the times [1]. Wang uses the dynamic relationship between roles and agents to describe the structure and behavior patterns of several agents and to resolve whether the representatives participating in the negotiation have formed a relatively stable cooperation. For official roles, the algorithm based on role exchange can be used for multiagent negotiation. This can improve the efficiency of negotiation between multiple agents in a data-poor environment. The test is carried out on the football simulation game platform. The results show that the algorithm is feasible and efficient. But this algorithm has some simulation limitations [2]. Hu provides an optimal trading model for representatives with a wide range of cases and game theory problems under the condition of less incomplete data. The computational complexity of Fatima's work and the size of the case library are controlled within a certain range. The results show that the negotiation model can be more useful and requires less time to make decisions. They provide a new possible algorithm for this model. The results show that the optimization performance of this model is better than other models. The business complexity algorithm requires a lot of manpower. It is difficult to achieve [3]. Blum first proposed the concept of graph planning. It was the first to use graphs to solve planning problems, providing people with a new idea when solving intelligent planning problems, which greatly enriched the field of intelligent planning.

In this paper, the agent's evolutionary algorithm is integrated into the actual problem—multi-issue autonegotiation of law. According to this problem, the autonegotiation solution process and corresponding model are designed. In addition, a new type of solution is proposed for multiple legal issues. Compared with traditional solutions, the applicability has great advantages. Among them, the autonegotiation result of all the agent's total utility can be

quickly found. In the changing environment, this article focuses on the multiagent system negotiation problem. According to the distributed information sharing of multiple agents, even if the case reveals incomplete information, the multiagent can be generated while ignoring the incomplete information. *Optimal Solution*. The experimental results show that the success rate of the system in analyzing multiple legal issues autonegotiation reaches 67.56% under the condition of incomplete information from the outside world. The MA-PDDL grammar is extended to allow the description of multiagents in a dynamic environment. The experimental content of simulation effectively shows that the method used in this paper is effective. In the context of multiple legal issues, a multiagent system can be used to solve problems and make consultation decisions.

## 2. Theoretical Analysis and Methods

*2.1. Multiagent Negotiation Problem.* Multiagent system (MAS) is an important research field of distributed artificial intelligence. Each agent in MAS has autonomy and distribution and can understand the needs of the owner, so it is a reasonable and effective method to apply the agent to autonegotiation. Multiagent cooperation means that in MAS, when the agent cannot perform the tasks assigned by the system due to insufficient capabilities or insufficient knowledge or resources, it needs to communicate with other agents of the current system to obtain the necessary solutions to the problem and cooperate with each other. In order to achieve the goal of solving problems together. The problem of multiagent cooperation is mainly to study how multiple agents in ADM work together to perform complex tasks, including how to communicate between agents and how to automate cooperation strategies. SMA is now widely used in all ages, and many practical problems cannot be solved by one agent. For example, in the manufacturing industry, to improve assembly efficiency, many robotic arms often work at the same time to assemble machines. In this case, between machines, cooperation is very important. In view of this, it can be said that the problem of multiagent cooperation is an urgent problem in MAS and a major problem in SMA research [4], distributed or centralized [5, 6]. Therefore, the basic structure is shown in Figure 1.

MAS is very flexible. System agents can perform certain tasks independently or cooperate with other agents to perform tasks as needed. Generally speaking, when encountering the following three situations, it may require collaboration between several representatives [7]:

- (1) Agents cannot perform the tasks assigned by the system. And there is another agent with problem-solving ability in the system, so the task is transferred to the agent with problem-solving ability to perform the task.
- (2) Some agents can perform tasks assigned by the system by themselves. But agent's knowledge shows that cooperating with other agents can improve workload and produce better results. At the same time, the representatives who hope to participate in

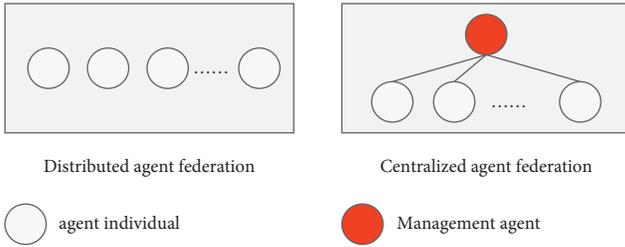


FIGURE 1: The basic structure of the agent federation.

the cooperation believe that cooperation will bring more benefits than the competition.

- (3) For some tasks, the agent can only partially complete, and other tasks must be completed by another agent.

When the system meets the above three conditions, there will be a need for collaboration. A complete collaboration process can be roughly divided into the following: (1) establishing a willingness to collaborate and setting collaboration goals, (2) planning model and participating collaborators, (3) recruiting agents who can participate in collaboration, (4) selecting specific collaboration strategies, (5) realizing common goals through collaboration, and (6) evaluating success in six steps.

Next, let us introduce Jiao Wenpin’s multiagent collaboration model of “positioning due to demand, striving for the position.” The collaboration model is defined as  $M = Ag, G, P, T, S$ , where we have the following:

- (1) Ag is a liaison, mainly based on the final task assigned by the system, and establishes a willingness to cooperate by establishing a collaborative goal. At the same time, he is responsible for selecting the most suitable partner, an agent that can participate in collaboration, that is, seeking the best solution.
- (2) G represents a set of cooperation goals, which is a set of simple cooperation goals created by the liaison officer based on the final goal.
- (3) P stands for collaborative planning; that is, after all the targets of G are assigned to related agents, these agents are scheduled, and the order of their operations is controlled.
- (4) T is a group of agents that work in collaboration with other agents, which is determined by the liaison officer according to the requirements of the  $Ag \in T$  collaboration target.
- (5) S stands for coordination strategy, that is, the coordination strategy of the relationship between agents participating in  $S \in P$  cooperation [8, 9].

The agent cooperation process is that once a new job is received, it first checks whether it is active and sends the work to the coordinating agent; otherwise, it matches its own capabilities, if it can be done independently; if the work cannot be done independently, then it is forwarded to the coordinating agent for cooperation. After receiving the work, the coordinator divides the task into several collaboration goals, then matches the capabilities of the agent in

the system, and selects the appropriate agent to complete the cooperation. The choice largely depends on the agent’s ability to complete the task and the time and resources required to complete the task when the coordinating agent receives multiple collaboration requests at the same time. This is a priority task, and the maximum value can be selected for the operation accordingly. The priority of the entire collaboration process is shown in Figure 2 [10].

**2.2. Multiagent Negotiation Mechanism.** After the above analysis of the multiagent negotiation problem, this section will develop a mechanism to solve this problem. With the rapid development of artificial intelligence and agent technology, transactions between legal persons, lawyers, and supplier networks have become more and more common. Online transactions have become an indispensable part of online transactions and are the most effective medium for both parties to transactions [11]. Therefore, the research of multiqualification negotiation with many domestic and foreign agent companies has reached a high level and continues to innovate. This section mainly studies and discusses the negotiation strategies of multiagent negotiation proposals. The counteroffer is in the process of negotiation. After the negotiating party proposed a plan, the other party felt that there was a deviation from the target’s expectations, and the remaining plan space of the other party had been adjusted. The new plan is redefined and a new strategy is proposed. The reduction mechanism is that the negotiating party considers the overall situation, revises the previous proposal, and reduces its own demand. It may be that the customer lowers the requirements for product features, or the supplier lowers the price of the product. The supplier allows the negotiation to continue. In the negotiation process, the trading strategy can be considered as a set of functions used to score specific trading issues at different points [12, 13]. We separate the unique semantic information of MA-PDDL from the standard PDDL semantics and call these two kinds of information at the same time during the planning and solving process to handle things that PDDL cannot handle. For example, when we deal with collaboration issues in the system, we first traverse the MA-PDDL file, use the string “action” for matching, and store the following string as a keyword in a Map, which is actually the entire system list of actions in.

Use  $propikm$  to view the quotations provided by  $agi$  to other agents that have been rejected. Use  $x[isikj]$  old to represent the  $isikj$  problem in the  $propikm$  proposition, and use  $Dikj$  to show the  $isikj$  range acceptable to  $agi$ . Use  $Vikj$  to display the scoring function of the middle element that interacts with  $isikj$ . This function assumes a monotonic increase or decrease. Use  $propikm + 1$  to indicate that  $agi$  is about to make a new offer. In this way, the negotiation strategy can be expressed as a work strategy:

$$\text{tactic: } Dikj \longrightarrow Dikj. \tag{1}$$

This function takes the old value of  $isikj$   $x[isikj]$  as input and returns the new value  $x[isikj]$  of problem  $isikj$  in  $agi$ ’s new proposition  $propikm + 1$ , and

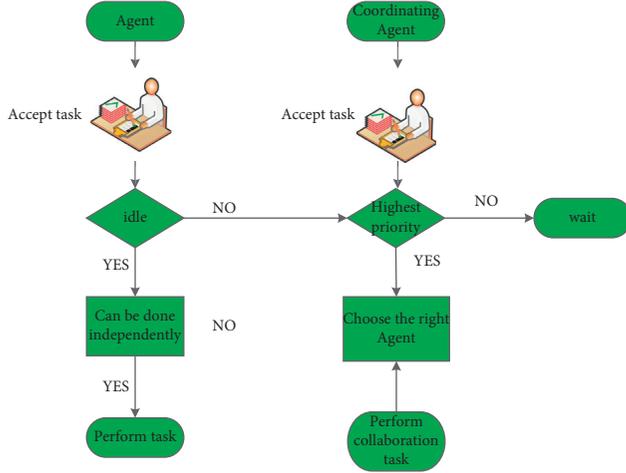


FIGURE 2: Negotiation flowchart.

$$Vikj(x[isikj]old) \geq Vikj(x[isikj]new). \quad (2)$$

The concession to the isikj problem can be defined as a decrease in expected profit, that is, a decrease in point value. At each stage of the negotiation process, consider five different negotiation and concession strategies for a single issue isikj:

- (1) No compromise: no compromise on the isikj problem, that is, the behavior does not modify the value of isikj
- (2) Small concessions: small concessions on the issue of isikj
- (3) Moderate concessions: give moderate concessions on the issue of isikj
- (4) More concessions: make more concessions on the isikj issue
- (5) Compromise and concession: the concession is completely isikj, that is, the new value  $x[isikj]$  of the problem isikj is the value of isikj proposed by the negotiating opponent

The new value  $x[isikj]$  new of the isikj problem can be calculated by the following formula:

$$x[isikj]new = x[isikj]old + (-1)wF |RVikj-x[isikj]old|. \quad (3)$$

When the Vikj function monotonously increases  $w = 0$ , it will be a parameter; when the Vikj function monotonously decreases,

$$w = 1, \quad F(F \in [0, 1]). \quad (4)$$

The equation guarantees

$$\begin{aligned} x[isikj]new &\geq RVikj, \\ x[isikj]new &\leq x[isikj]old. \end{aligned} \quad (5)$$

In other words, once the value of isikj has been raised, this value cannot be raised again. This is the basic standard

for negotiation. Otherwise, it tends to fall into an infinite loop.

The parameter  $F$  can be a constant. By assigning different values to  $F$ , five different concession strategies can be described as follows.

Showing no concessions:

$$F = 0. \quad (6)$$

Expressing a slight concession:

$$F \in (0, 0.5), \quad (7)$$

Indicating a moderate concession:

$$F = 0.5. \quad (8)$$

Expressing a major concession:

$$F \in (0.5, 1). \quad (9)$$

Expressing compromise:

$$F = 1 \text{ 或 } F = \left| \frac{(x[isnkj]-x[isikj]old)}{(RVikj-x[isikj]old)} \right|, \quad (10)$$

where  $F$  can be a function of a variable, or a standard function, which represents all the concessions made by the actor to the problem isikj. Based on this, the following two functions can be used to represent  $F$ :

$$F = 1 - \gamma e^{-\left| \frac{C}{Ctotal} \right|}, \quad \gamma \in R+, \quad (11)$$

$$F = \gamma e^{-\left| \frac{1-C}{Ctotal} \right|}, \quad \gamma \in R+.$$

These two functions are used to describe the general monotonic concession reduction strategy. As traders' demand value gets closer and closer to the bottom-line value, discounts are getting smaller and smaller, so the initial demand curve will drop sharply. But as time passed, the negotiation gradually proceeded. The bottom-line value method is shown in Figure 3.

Combining five different concession strategies in the same strategy model is very useful for the development and programming of actual application systems: users only need to specify the value of parameter  $F$  or  $\gamma$  to define it, thereby increasing the independence and customization of system modules [14, 15]. Each agent has its own negotiation strategy, but these strategies are only predetermined by the user, and only we consider the user's unilateral interests and need to negotiate to achieve the most favorable situation that users want. However, when multiple agents coexist in the same environment, each agent only considers the interests of its users. Therefore, it is difficult to reach a consensus through negotiation, so it is necessary to establish a concession rejection mechanism. First, the agent configured in the multiagent negotiation system is defined as

$$C = \{b, M\}, \quad (12)$$

where  $b$  represents the client participating in the negotiation.

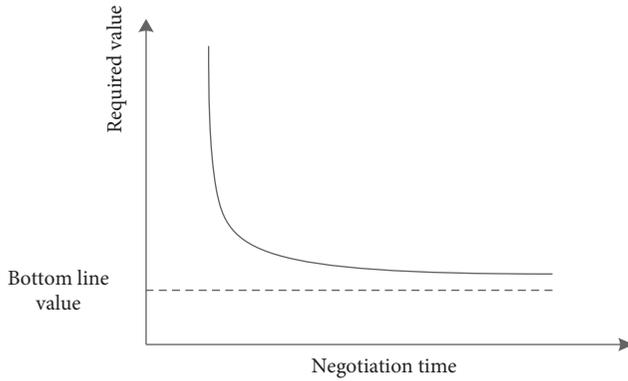


FIGURE 3: Approximation of the required value to the bottom line in the monotonic reduction concession strategy.

$M = \{m_1, m_2, \dots, m_n\}$ , where  $M$  represents a supplier representative who may have a negotiated relationship with a customer representative,  $m_i$  represents a unit representing a single supplier, and  $M$  includes a supplier representative.

Then, we define an agent solution repository, which represents the operations and methods that each agent can perform during the negotiation process. The collection is

$$S(X) = \{sx_1, sx_2, \dots, sx_n\}. \quad (13)$$

Among them, the strategy is in  $S(X)$  which is related and conflicting. Finally, the evaluation utility function of  $\text{AgentR}(y)$  is defined to evaluate the propositions in the strategy library. Therefore, we can show the negotiation strategy between the customer representative and the supplier representative of the transaction:

$$P = \langle C, S, R \rangle. \quad (14)$$

The franchise quotation negotiation mechanism is used to resolve  $P$ , that is, a mechanism for negotiation strategy conflicts between users. The negotiation mechanism generally includes three important links:

- (1) Customer representatives and supplier representatives evaluate public services
- (2) Provided by customer representatives or supplier representatives
- (3) Customer representative proposal or supplier representative proposal

The relationship of the three negotiations links between customer representatives and supplier representatives is shown in Figure 4.

**2.3. Demand Analysis of Multiattribute Utility Autonegotiation Model.** A multiattribute negotiation proposal usually consists of several attributes: the necessary attributes are the price, the brand of the refrigerator, the capacity of the refrigerator, the height of the refrigerator, the power consumption of the refrigerator, the warranty of the refrigerator, and additional services, etc. Customers must first clarify these when choosing to purchase specific

requirements for features, such as expected price, Haier brand, 2 m high capacity, low power consumption, and more home delivery [16]. How to evaluate the customer's quotation or how to evaluate the customer's attention to each function evaluation method is necessary; this evaluation method can be completed by using a single attribute of the product. Or it can comprehensively evaluate all the characteristics of the product requested by the customer. Therefore, the multiattribute negotiation model should have the following characteristics:

- (1) There is a clear reasoning mechanism for trading strategies and trading processes.
- (2) Perform utility evaluation based on the overall and individual characteristics of the proposal.
- (3) Create a new quotation based on the benefits of an attribute and the benefits of the owner. The multiattribute negotiation model has the functional modular structure of the modules in the negotiation model. The model can be divided into five parts: the proposed utility definition module, the attribute planning module, the negotiation function evaluation module, the coordination module, and the utility planning module. The mutual calling relationship between them is shown in Figure 5.

The main task of the utility planning module is to understand the needs of users and formulate a general plan for the utility of the product based on the current negotiated learning experience. The attribute planning module is based on the utility planning module to perform partial analysis of product attributes. The main task of the attribute evaluation module is to obtain quotations from users and then execute the utility function to evaluate the benefits of each attribute of the item in the quotation. The usefulness of offer: most of the evaluation modules of the distribution module are compiled by the attribute evaluation module. It is recommended to consider the willingness of the element through the weight of each attribute. The negotiation and coordination module is used to coordinate the negotiation, modeling, and analysis of the negotiation status of each module and strengthen the control of the negotiation between these states.

The main points that the negotiation and coordination module needs to pay attention to when analyzing the negotiation status are as follows:

- (1) No matter whether the same negotiation process is always going on during the negotiation process; that is, the negotiation usually enters a monotonous cycle, indicating that the two parties have settled. The characteristics and advantages of the products offered by refusal to concession have not changed. It can be seen that the negotiation has reached a stagnation stage. If the negotiation time exceeds the negotiation time, the negotiation and coordination module will forcibly stop the negotiation.
- (2) In the negotiation party's proposal, if the utility supply of certain characteristics of the same product

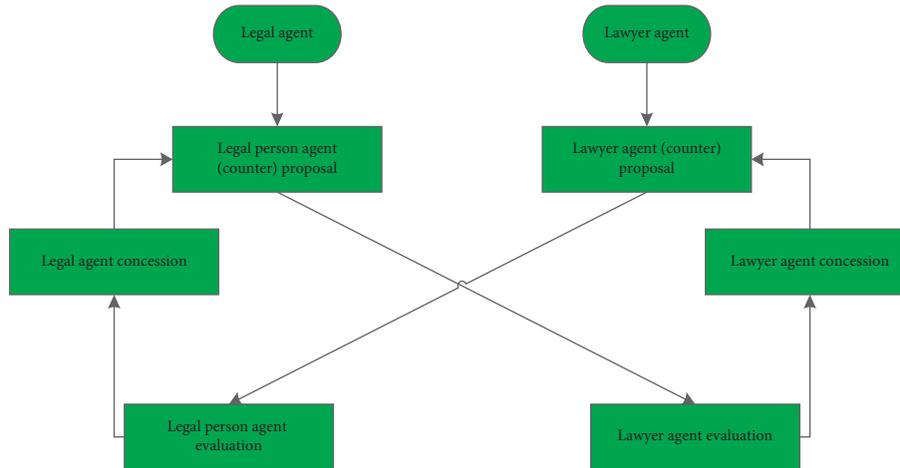


FIGURE 4: Concession counterproposal negotiation mechanism of multiagent negotiation.

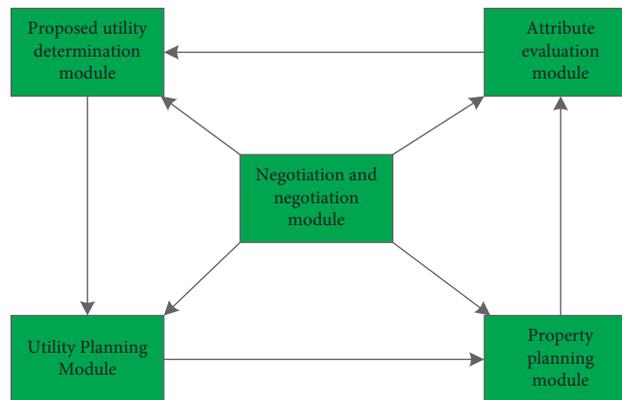


FIGURE 5: Process model of multiattribute negotiation.

differs too much, the greater the difference, the higher the negotiator's satisfaction with this attribute. Because it is only an attribute element, the coordination module only needs to negotiate and adjust this attribute.

- (3) Regardless of whether the overall utility proposal of the same product differs too much from the proposal of the negotiating party, the current negotiation may be stalled. And a coordination module is needed to coordinate the negotiation parties to see the overall situation [17, 18].

### 3. Experimental Simulation

**3.1. Scene Simulation Module.** The main function of the scene simulation module is to display the system environment graphically. The use of Java drawing technology mainly includes drawing the world scene and drawing various objects in the system. We display the entire world scene in a grid format. During initialization, the canvas size is set to a rectangular world of  $500 \times 500$  pixels. The size of the stage is determined by the length and width entered by the user. Assuming that the size of the scene input by the user is  $M \times N$ , the entire world is drawn sequentially as an  $M \times N$  grid, and

each grid corresponds to its coordinates. Its specific purpose is to call the method of drawing lines in the graph. First, we draw a vertical line  $M$  on the  $X$  axis. The pixels between the lines are calculated by dividing all the pixels of the length of the world by  $M$ , and then, we draw  $N$  on the  $Y$  axis. The pixels between the lines are between the horizontal lines and the lines. The pixels are calculated by dividing all the pixels of the world width by  $N$ . In this way, the entire world has created a coordinate system with the  $X$  axis from 0 to  $M$  and the  $Y$  axis from 0 to  $N$ . When we describe other objects in the scene, we only describe them in this coordinate system without considering specific pixels [19].

Algorithm 1 describes the entire process of continuous planning. There are mainly two subprocesses, which, respectively, represent the planning process and the execution monitoring process, which are implemented by calling Algorithm 2 and Algorithm 3. After entering the current state of the system and the desired target state, we first determine whether the target state is reached; if it has been reached, return true; if not reached, call Algorithm 2 to solve a planning solution; if the planning solution is empty, it means that the target state cannot be reached. If reached, return false; otherwise, call Algorithm 3 to execute the planning solution. After the execution, continue to judge whether the

planning solution is obtained, and so on, until the 27 tasks are completed or the task is found to be unable to be completed.

Under normal circumstances, a system can consist of multiple objects. For different objects, we use different colored geometric figures instead. In more complex situations, when drawing system objects, you need to add text to the geometry to distinguish different objects. We only need to know the coordinates of the object. For objects with multiple coordinates, such as parts, all coordinates covered by the object should be defined and drawn in the same shape. Figure 6 draws a simple model. It is mainly composed of four types of objects: a room, a corridor, a table, and two AGVs. We divide the whole world into a 10\*10 grid. The white part is the part that the AGV can traverse. We represent the corridors in the system. The room is the part that the agent cannot directly pass through. It can be seen as an obstacle, represented by a black square, table, and AGV, and represented by a circle and text, respectively. In this scene, we can clearly see the location of each room and the AGV and grille. According to this position data, we can simulate the movement of the AGV from the current position in Figure 6 [20, 21].

This picture is a 10\*10 microcosm of the experimental simulation. There are one hundred experimental simulation data in the experimental simulation. According to the negotiation model proposed in this research, the agent will start the process of negotiating the problem  $u$  before the multiagent system provides the negotiation description. The negotiation process and the meaning of the negotiation subject imperative sentence are shown in Table 1.

The calculation results of the agent's individual trust are shown in Table 2. Finally,  $T=0.9885$ ,  $R=0.8412$ , and  $C=0.9145$ .

**3.2. Negotiation Model.** Negotiations can be divided into three types according to the number of representatives participating in the negotiation and the number of negotiation issues: two-way unilateral negotiation, multiobjective bilateral negotiation, and multipurpose multiparty negotiation. The negotiation framework of multilateral and multiobjective negotiation that supports multiple rounds of negotiation is shown below. First, some assumptions have been made. The following discussion is based on these assumptions:

Suppose 1 agent person is interested in himself, which means he seeks the best in himself.

Assumption 2: representatives have a series of reasons.

Assumption 3 represents incomplete information.

Prerequisite 4: both parties are sincere and hope to reach an agreement through negotiation. In other words, there is no fraud.

Hypothesis 5: the worst result of the negotiation is the failure of the agreement, that is, the failure of the negotiation [22].

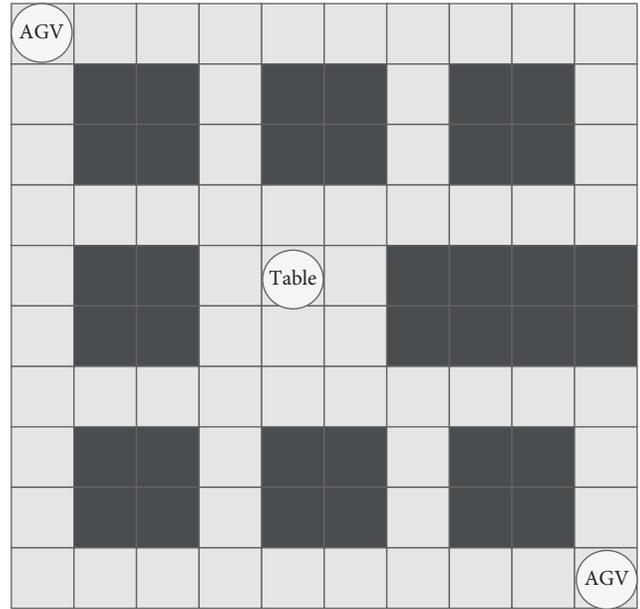


FIGURE 6: A simple simulation scenario.

TABLE 1: The meaning of symbols in the negotiation process.

Negotiation	Negotiation in multiagent system
Process	The agent initiates a negotiation process about problem $u$
$U$	The problem $u$ is raised by the agent
Thread	Agent negotiation on issues
Proposer	The agent that initiated the negotiation
Participator	Respond to the set negotiated by the agent

TABLE 2: Agent's interaction history data.

Provider Agent	S number	U number	Total number	City of provider
Izsws	2013	9	2156	Shenzh
Jojphp	2459	5	2435	Shenzh
Yykele	14256	70	14569	Shenzh
Ayalawang	7159	3	7158	Shanghai

Here, caution refers to behavior that should increase his interest by talking. Rather than a true practical decision theory, it uses optimal development as the criterion for rational behavior. Ethics uses all social exploitation as the norm for rational behavior. Conversation theory is not sufficient to judge the rational behavior of negotiation participants. This largely depends on the other participants in the negotiation. The strategies and tactics that negotiators can use to maximize their advantages are the rational behavior of the negotiators in the negotiation process. Finding other participants in a conversation with most recipients is also considered a form of conversation. Figure 7 briefly describes the process.

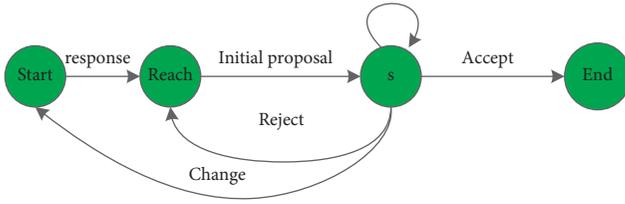


FIGURE 7: Negotiation process.

3.3. *Three-Dimensional Space Preference Negotiation.* The use of the hypercube proposed by Ito et al. is to express the preference of a subject through a series of space-based three-dimensional hypercubes. It uses an  $N+1$  hypercube to represent a set of utility values with  $N$  link points (for example, as shown in Figure 8); the edges of the hypercube in the same dimension represent the problem and the range of values. As a complex nonlinear utility space, the hypercube utility will express the object's preference in a more concrete and realistic way. But it also makes the solution search area very large [23, 24].

When resolving the negotiation results, Ito et al. used a virtual fusion algorithm to find a suitable solution in an area not lower than the agent's reserved utility, as a representative set of candidate proposals. Then apply the adaptive system. Time-consuming bidding and concession strategies with limited proposal mechanisms and opponents take turns to find the final solution [25, 26]. The agreement greatly reduces the search area, while ensuring the authenticity of the Pareto transaction results and greatly improving the search performance. Bracewell expands the utility of hypercubes based on Ito's research. It is suitable for some hypercubes, and its usefulness can be constant. The linear function or sine function summarized above is not difficult to see that no agreement can be applied to all transaction situations, and the change and openness of the negotiation situation have also become a problem and focus of the design of the negotiation protocol [27].

## 4. Experimental Results and Data Analysis

4.1. *Negotiation Time Analysis.* Obviously, the experiment process takes a much longer time for man-machine negotiation. The reason for the analysis comes from one reason. Different people have different abilities and different life paths. Moreover, the level of negotiation expertise is different, which can cause minor misunderstandings, minor mistakes, etc. and waste time. Secondly, the processing power of humans is still far behind; it is smaller than the processing power of computers, so it takes more time to calculate the similarity. The utility calculates and generates the best return. In the final analysis, the round of negotiation used by humans and machines to reach an agreement is less than the round of negotiation used by everyone to reach an agreement. Figure 9 compares the advantages of  $H_{can}$ ,  $H_{Emp}$ , and  $Q_{Emp}$ .

Negotiations between people are obviously more time-consuming than negotiations between humans and machines. The analysis is due to one of the reasons. Different

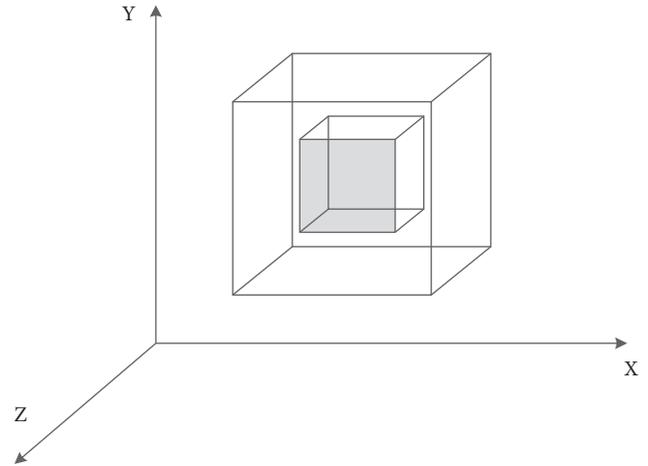
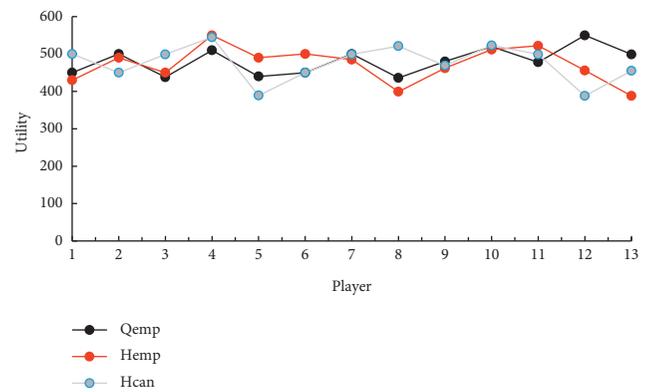


FIGURE 8: Examples of the utility of hypercubes.

FIGURE 9: Comparison of experiment utility among  $H_{can}$ ,  $H_{Emp}$ , and  $Q_{Emp}$ .

people have different ability levels and life backgrounds, and the degree of mastery of negotiation essentials is different, which will cause small misunderstandings, small mistakes, etc., and waste a lot of time. Secondly, people's computing power is far greater. It is lower than the computing power of the computer, so it takes more time in similarity calculation, utility calculation, and optimal return price generation. The reason for the analysis is that since the legal entity has a large number of cases in advance and has a large amount of experience and knowledge, it can choose the best transaction, which will ensure that the average utility of the seller increases. And because the opponent's acceptance is more of the opponent's consideration, it is easy to admit that negotiation may reach an agreement faster. This will greatly speed up the negotiation process and finally reach an agreement and can greatly improve the efficiency of the negotiation. If you increase the number of trading cycles between the two sides and increase the size of the case library, it can be seen that in essence, it only increases the polynomial time, with little effect. Therefore, the process of simulating legal multi-issue autonegotiation in this experiment has been substantially improved, in terms of both time and accuracy; these performances are reflected in the results

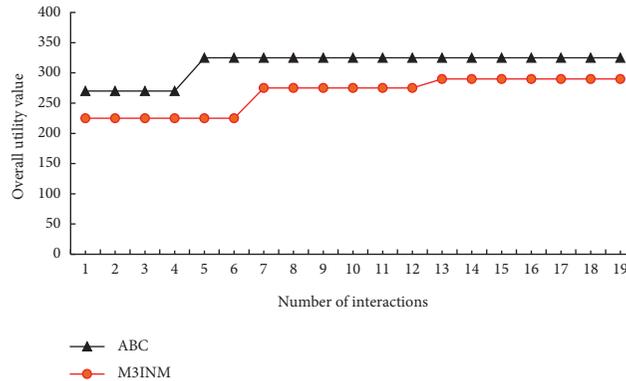


FIGURE 10: Convergence performance comparison.

to a certain extent. There was no significant increase in time during the experiment [28].

**4.2. Iterative Analysis of Negotiation Algorithm.** Using the modeling method in this article, from the weight matrix  $W$ , the utility value matrix  $V$ , and the utility adjustment range  $\{V_{\min}, V_{\max}\}$ , the main parameters are defined as the maximum range of the attribute parameter  $P = 10$ , which is 2N freshwater resources ( $= 8$ ), the number of bees,  $X = 3$ , and the number of attributes  $M = 3$  and  $r = 2$ . Figure 10 is the convergence curve of the artificial bee colony algorithm model. It can be seen from the figure that the best solution is not modified after 20 cycles of the algorithm. In the four sets of experiments, through the console area under the tool, we can see which actions each agent in the system performed during the whole process. From the console output results of experiments a and c, it can be seen that when using MA-PDDL for description, no matter whether the system environment changes or not; it is only planned once and executed according to the planning solution. The average time of each run is 1.8430 seconds, and the overall maximum utility value is 322.922 (reserved, decimal point) (the last 3 digits). At this time, the entity should select the agent person RB, select Q and C, and select W, the current utility value of all the agent persons, and the attributes in the multiple agents and objects (M3INM) negotiation method given in the Object = 1 document. The problem described by the model is similar to this article. The model of this negotiation method is one of the autonegotiation methods, but this negotiation method has certain limitations because it only stays in a single agent. In order to verify the performance of the model in this article, we use the M3INM model to troubleshoot the example calculations in this article. The results are compared with the results of the ABC algorithm in this paper. One iteration of the model takes 1.84 seconds, and the best solution for 20 iterations is 322.922. M3INM simulation requires 3.254 seconds, and the best solution for 20 iterations is 321.958. This model and the M3INM model each have 50 operations. And the results show that the calculation result and correction speed of the algorithm in this paper are better than the conventional algorithm. Figure 10 is a comparison diagram of the solution results of the ABC algorithm and the M3INM model. It can be seen from the figure that the performance of the solution of the

ABC algorithm is better than that of the M3INM, and it can converge to the best solution in the world. In 10 negotiations, the overall practical value of M3INM reached the 20th generation. All are below 300.

## 5. Conclusion

In this paper, the agent's evolutionary algorithm is integrated into the actual problem—multi-issue autonegotiation of law. According to this problem, the autonegotiation solution process and corresponding model are designed. In addition, a new type of solution is proposed for multiple legal issues. Compared with traditional solutions, the applicability has great advantages. Among them, the autonegotiation result of all the agent's total utility can be quickly found. In the changing environment, this article focuses on the multiagent system negotiation problem. According to the distributed information sharing of multiple agents, even if the case reveals incomplete information, the multiagent can be generated while ignoring the incomplete information. Optimal Solution. The experimental results show that the system has a relatively high success in analyzing multi-issue automatic negotiation under the condition of incomplete information from the outside world. The more negotiation experience and background knowledge you have, the better the negotiation results will be achieved. In addition, this paper has achieved 67.56% of the resolution rate of multi-issue legal autonegotiation through experimental simulation, which is a very good experimental result for multi-issue legal autonegotiation. The work in this chapter focuses on the strategy and method for generating optimal solutions under the condition of incomplete information, which has been mostly ignored in the past, and the experimental results verify the negotiation efficiency of the method under the condition of incomplete information. There is still much work to be done for case-based multitopic negotiation. How to choose the most suitable case is an important issue it faces. Therefore, the selection and improvement of the similarity function must be further deepened. In addition, the reuse of cases is also a meaningful work. How to make past cases suitable for the present situation is a content worthy of careful study. Generally speaking, there are two directions that can be further deepened in the future. One is that

Bayesian classification algorithms can be constructed to deal with incomplete information, and the second can be extended to larger databases and application environments.

## Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

## Conflicts of Interest

The authors declare no conflicts of interest with respect to the research, authorship, and/or publication of this article.

## Acknowledgments

This work was supported by the National Social Sciences Fund of China (no. 20CFX006) and the Research Fund of Jimei University (Grant no. Q202112).

## References

- [1] N. Bulling, "Special issue on multi-agent systems," *Annals of Mathematics and Artificial Intelligence*, vol. 78, no. 1, pp. 1-2, 2016.
- [2] Z. Wang and H. Zhang, "Observer-based robust consensus control for multi-agent systems with noises," *Neurocomputing*, vol. 207, pp. 408–415, 2016.
- [3] W. Hu, L. Liu, and G. Feng, "Consensus of linear multi-agent systems by distributed event-triggered strategy," *IEEE Transactions on Cybernetics*, vol. 46, no. 1, pp. 148–157, 2017.
- [4] L. I. Chang-Jiang and G. P. Liu, "Data-Driven leader-follower output synchronization for networked non-linear multi-agent systems with switching topology and time-varying delays," *Journal of Systems Science and Complexity*, vol. 31, no. 1, pp. 87–102, 2018.
- [5] M. Elhoseny, "Multi-object detection and tracking (MODT) machine learning model for real-time video surveillance systems," *Circuits, Systems, and Signal Processing*, vol. 39, pp. 611–630, 2019.
- [6] L. T. J. Munwar, A.-A. Abdel-Haleem, M. Y. Abubakar, M. Elhoseny, and I. Ali, "Semantic-k-NN algorithm: an enhanced version of traditional k-NN algorithm," *Expert Systems Applications*, vol. 151, Article ID 113374, 2020.
- [7] M. Jie, R. Wei, and C. Jie, "Distributed consensus of second-order multi-agent systems with heterogeneous unknown inertias and control gains under a directed graph," *IEEE Transactions on Automatic Control*, vol. 61, no. 8, pp. 2019–2034, 2016.
- [8] Z. Wang, W. Wang, and H. Zhang, "Robust consensus for linear multi-agent systems with noises," *IET Control Theory & Applications*, vol. 10, no. 17, pp. 2348–2356, 2016.
- [9] L. Wei, W. Gu, and W. Sheng, "Decentralized multi-agent system-based cooperative frequency control for autonomous microgrids with communication constraints," *IEEE Transactions on Sustainable Energy*, vol. 5, no. 2, pp. 446–456, 2017.
- [10] S. Wang, J. Wan, D. Zhang, D. Li, and C. Zhang, "Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination," *Computer Networks*, vol. 101, pp. 158–168, 2016.
- [11] Q. Wang and P. Lu, "Research on application of artificial intelligence in computer network technology," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 33, no. 5, Article ID 1959015, 2019.
- [12] W. Hu, L. Lu, and F. Gang, "Output consensus of heterogeneous linear multi-agent systems by distributed event-triggered/self-triggered strategy," *IEEE Transactions on Cybernetics*, vol. 47, no. 99, pp. 1914–1924, 2017.
- [13] C. J. Li and G. P. Liu, "Data-driven consensus for non-linear networked multi-agent systems with switching topology and time-varying delays," *IET Control Theory & Applications*, vol. 12, no. 12, pp. 1773–1779, 2018.
- [14] Y. Zheng, J. Ma, and L. Wang, "Consensus of switched multi-agent systems," *IEEE Transactions on Circuits & Systems II Express Briefs*, vol. 63, no. 3, pp. 314–318, 2017.
- [15] Y. Jiang and J. C. Jiang, "Diffusion in social networks: a multiagent perspective," *IEEE Transactions on Systems Man & Cybernetics Systems*, vol. 45, no. 2, pp. 198–213, 2017.
- [16] L. Han, X. Dong, Q. Li, and Z. Ren, "Formation-containment control for second-order multi-agent systems with time-varying delays," *Neurocomputing*, vol. 218, pp. 439–447, 2016.
- [17] C.-J. Li and G.-P. Liu, "Consensus for heterogeneous networked multi-agent systems with switching topology and time-varying delays," *Journal of the Franklin Institute*, vol. 355, no. 10, pp. 4198–4217, 2018.
- [18] M. R. Basir Khan, R. Jidin, and J. Pasupuleti, "Multi-agent based distributed control architecture for microgrid energy management and optimization," *Energy Conversion and Management*, vol. 112, pp. 288–307, 2016.
- [19] Y. Yan and J. Huang, "Cooperative output regulation of discrete-time linear time-delay multi-agent systems under switching network," *Neurocomputing*, vol. 241, pp. 108–114, 2017.
- [20] G. Wen, Y. Zhao, Z. Duan, W. Yu, and G. Chen, "Containment of higher-order multi-leader multi-agent systems: a dynamic output approach," *IEEE Transactions on Automatic Control*, vol. 61, no. 4, pp. 1135–1140, 2016.
- [21] D. Weyns, H. Parunak, and F. Michel, "Environments for multi-agent systems III," *The Knowledge Engineering Review*, vol. 3374, no. 4, pp. 409–421, 2016.
- [22] C. Long, Y. Wang, and R. Wei, "On convergence rate of leader-following consensus of linear multi-agent systems with communication noises," *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3586–3592, 2016.
- [23] M. Challenger, G. Kardas, and B. Tekinerdogan, "A systematic approach to evaluating domain-specific modeling language environments for multi-agent systems," *Software Quality Journal*, vol. 24, no. 3, pp. 755–795, 2016.
- [24] S. Bolognani, R. Carli, E. Lovisari, and S. Zampieri, "A randomized linear algorithm for clock synchronization in multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 61, no. 7, pp. 1711–1726, 2016.
- [25] J. Xi, Y. Yu, and G. Liu, "Guaranteed-cost consensus for singular multi-agent systems with switching topologies," *IEEE Transactions on Circuits & Systems I Regular Papers*, vol. 61, no. 5, pp. 1531–1542, 2017.
- [26] S. Chen, D. Ho, and L. Li, "Fault-tolerant consensus of multi-agent system with distributed adaptive protocol," *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2142–2155, 2017.
- [27] Y. Zheng, J. Ma, and L. Wang, "Consensus of hybrid multi-agent systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 4, pp. 1359–1365, 2018.
- [28] V. H. Bui, A. Hussain, and H. M. Kim, "A multiagent-based hierarchical energy management strategy for multi-microgrids considering adjustable power and demand response," *IEEE Transactions on Smart Grid*, vol. 9, no. 99, pp. 1323–1333, 2016.